

基于 DM642 PCI Master 读方式的 AVS 视频数据传输实现

秦永红¹, 戴小文²

(1. 西南交通大学 电气工程学院, 四川 成都 610031;
2. 厦门大学 机电与物理学院, 福建 厦门 361005)

摘 要:介绍了 AVS 解码器的系统构成;研究了基于 TMS320DM642 芯片实现 PCI 主模式读方式的数据传输系统。给出了在嵌入 Linux 操作系统下通过 DM642 PCI 主模式读方式实现 AVS 视频数据传输的具体方法;给出了视频数据接收线程与视频解码线程间的相互切换方法并配合内存映射、环形缓冲等方法,实现 AVS 视频流实时解码。实际结果证明此解码系统能满足实时性要求。

关键词: 数据传输; AVS; PCI 主模式; TMS320DM642; Linux

中图分类号: TN919.81 文献标识码: A 文章编号: 1673- 629X(2008)05- 0185- 03

Design of AVS Data Stream Transfer Subsystem Based on TMS320DM642 PCI Master

Q IN Yong hong¹, DAI Xiao-wen²

(1. College of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China;
2. College of Physics of Xiamen University, Xiamen 361005, China)

Abstract: The AVS decoder system is provided. The data transfer subsystem based on TMS320DM642 PCI master mode is designed. The method in the Linux to transfer AVS data stream between the host and target is presented. The switch way between the data stream transfer thread and AVS decoder thread is given. Meanwhile, the memory mapped and round buffer is used in data transfer subsystem. The AVS decoder in real time is fulfilled. The test results indicated that this subsystem can be used in real time.

Key words: data stream transfer; AVS; PCI master mode; TMS320DM642; Linux

0 引 言

AVS 的出现使得全球范围内可选的新一代视频标准形成了三足鼎立的局面: 国际标准 MPEG- 4、H. 264 和具有中国自主知识产权的 AVS 国家标准。

传统的方法采取主机向 DM642 子系统的存储器拷贝数据, DM642 再从存储器中取出数据进行编解码,但速率难以满足实时编解码的要求。笔者采用在主机主存中分配存储空间,在 DM642 中用 PCI 主模式方式与主机主存进行高速数据交换。

1 DM642 PCI 的性能介绍

DM642 为 TI 公司的高性能多媒体处理器,主频

收稿日期: 2007- 08- 22

作者简介: 秦永红 (1976-), 男, 湖南人, 硕士, 软件架构师, 研究方向为视频通信; 戴小文, 博士, 副教授, 研究方向为数字图像处理。

为 720MHz, 处理速度为 5760MIPS。PCI 接口支持 PCI2. 2 版本, 具有主模式/ 从模式接口, 32bit 地址/ 数据总线, 66MHz 时钟。主机通过 PCI 接口可以访问 DM642 整个片内 RAM、集成外设以及片外存储器^[1]。主模式访问突发数据长度最大为 64k。

DM642 支持 4 种类型的 PCI 数据交换: 从模式写、从模式读、主模式写以及主模式读^[1]。DM642 的 PCI 接口中包括三类寄存器: PCI 配置寄存器、PCI I/O 寄存器以及映射在 DM642 存储空间, 作为外设的 PCI 寄存器^[2]。

DM642 PCI 接口为 PCI 主机访问 DM642 的存储空间提供了两种方式: 可预取的数据访问和不可预取的数据访问。PCI 接口的主模式需要 DM642 控制, 相关的控制寄存器有: DM642 主地址寄存器 (DSPMA)、PCI 主地址寄存器 (PCIMA)、PCI 主模式控制寄存器 (PCIMC)^[3]。

2 DM642 PCI Master 读方式的优点

数据传输使用以 DM642 为主的方式,主要是因为如果使用以主机(ASIC)为主的 PCI 传输,在每次完成数据传输时会在 DM642 上触发传输完成中断,而在主机上并没有相关信息,因此在数据传输完成 DM642 收到中断时还要给主机发送一次传输完成的消息,这样会造成系统结构的复杂性,同时影响 PCI 传输的响应速度。使用以 DM642 为主的 PCI 传输会遇到两个问题:

- (1) ASIC 的内存管理采用了分页的方式,用户空间的连续内存,实际上可能是由不连续的物理内存页组成的,所以数据必须分页传输^[4]。
- (2) DM642 的 PCI 传输最大传输数据包长度为 65535 字节,超过这个大小的数据包要分多次传输^[5]。在 Host 端的 DM642 PCI 驱动程序中实现了一个内存映射的功能,将 Host 机(ASIC)核心内存映射到 ASIC 用户空间,并将该段内存的分页信息写入 DM642 的内存空间。在 DM642 的 PCI 接口软件模块中,设计了一个 PCI 传输请求队列,来实现提交传输申请和实际传输的异步执行机制,传输完成通过一个信号量对象来传递。对于每一个传输请求,都会以 ASIC 的内存页为单位传输,ASIC 内存分页信息通过 ASIC 上的 DM642 PCI 驱动程序写入 DM642 内存空间,保证传输的正确性。

3 DM642 PCI Master 方式数据传输的实现

Host 机(ASIC)与 Target 机(DM642)之间通过 PCI 总线传输码流和控制数据系统示意图如图 1 所示。

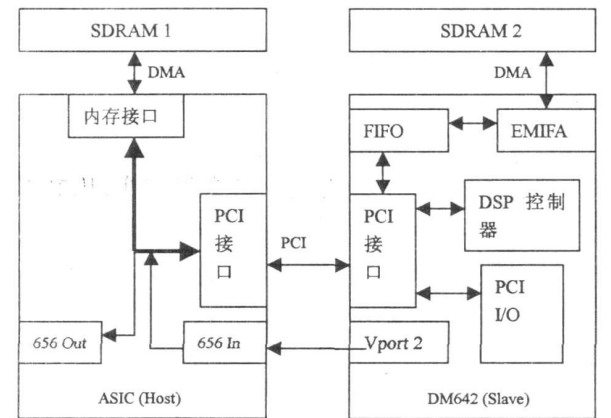


图 1 解码器系统框架图

Host 机将视频数据传输给 DM642 的过程如下: Host 机上的用户程序在启动时调用 PCI 驱动提供的创建内核缓冲区的功能创建内核缓冲区,并获得在用户空间中的地址。Host 将码流数据拷贝到内核缓冲

区中,然后发送一条包含码流类型、在内核缓冲区中的起始位置和码流长度的消息给 DM642;当 DM642 接收到此消息后,从消息中得到码流类型和长度信息,然后启动 PCI 传输,将 Host 内核缓冲区中的码流传输到 DM642 的内存中。DM642 在接收到视频码流之后,根据消息中携带的码流类型启动对应的解码操作。

4 DM642 程序结构

DM642 程序由消息接收线程和视频解码线程组成。

- (1) 消息接收线程:此线程中包含一个消息处理循环,不停地接收消息,并进行相应的处理。根据视频 AVS 数据到达消息,启动 PCI 传输,将 AVS 视频数据传输到环形缓冲区中,等待视频解码线程将其取走。

对 PCI 的配置如表 1 所示^[2,6]。

表 1 DM642 PCI 寄存器配置表

PCI 寄存器	发送(写)	接收(读)
dspma	srcAddr(源地址)	dstA ddr
pcima	dstAddr(目的地址)	srcAddr
pcime	byteCnt(字节数)+ options	byteCnt+ options
trctl	PCI EDMA 配置	

码流传输函数的具体代码如下:

```
static Void doTransfer(C64XX_PCL_Request * request)
{
    if(xfrMode= PCL_WRITE)
    {
        config.dspma= (Uns)req->srcAddr;
        config.pcima= (Uns)req->dstAddr;
        config.pcime= GET_BYTE_COUNT((req->byteCnt));
    }
    else if( xfrMode= PCL_READ_PREF || xfrMode= PCL_READ_NOPREF)
    {
        config.dspma= (Uns)req->dstAddr;
        config.pcima= (Uns)req->srcA ddr;
        config.pcime= GET_BYTE_COUNT( req->byteCnt);
    }
    SEM_PCL xfrConfig(& config);
    PCL xfrStart(xfrMode);
}
```

- (2) 视频解码线程:此线程包含一个解码循环,不停地从视频数据环形缓冲区中取出 AVS 视频数据、解码,然后从 Vport2 口输出。

5 实验结果

测试的目标板结构图如 1 所示。ASIC 连接的

flash memory 中已写好 Linux 系统。另有一台 PC 用来提供 NTFS 文件服务。将目标板的视频输出接到 PC 的视频采集卡的 Video In 接口。

AVS 解码测试的数如下所示:

frame 0 takes: 19.66 msl | frame length is: 2402 bytes.
frame 1 takes: 19.31 msl | frame length is: 2323 bytes.
frame 2 takes: 19.73 msl | frame length is: 2373 bytes.
frame 3 takes: 19.10 msl | frame length is: 2207 bytes.
frame 4 takes: 19.47 msl | frame length is: 2466 bytes.
frame 5 takes: 20.28 msl | frame length is: 2520 bytes.
average time per frame is: 19.59 ms

AVS 码流解码后播放的图像如图 2 所示。

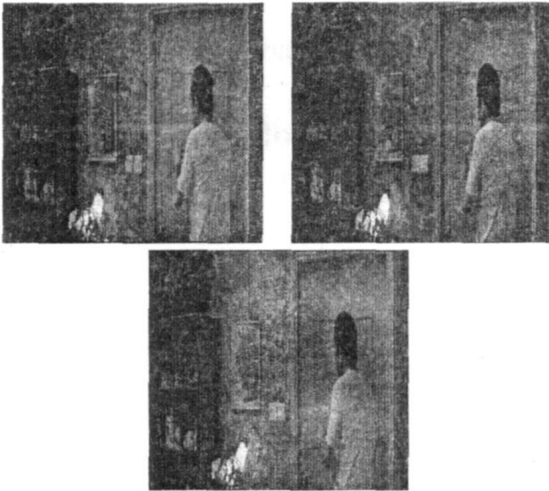


图 2 对 AVS 码流播放效果图

通过使用 PCI 主模式读的方式对视频数据流进行传输, 完全能满足解码要求, 且经过 AVS 解码器解码

后取得了良好的播放效果。

6 结 语

介绍了基于 DM 642 的视频解码器处理系统。采用了 DM 642 的 PCI 主模式读方式的 DMA 数传输, 实际应用证明该方案能够实现 AVS 视频流的实时解码后正确播放。

参考文献:

[1] Texas Instruments. TMS320C6000 Peripherals Reference Guide Literature Number: SPRU190D[M]. Texas: Texas Instruments Incorporated, 2001.

[2] Texas Instruments. TMS320C6000 DSP/ BIOS 5.31 Application Programming Interface (API) Reference Guide Literature Number: SPRU 403N[M]. Texas: Texas Instruments Incorporated, 2006.

[3] Texas Instruments. TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide Literature Number: SPRU 234[M]. Texas: Texas Instruments Incorporated, 2003.

[4] 毛德操, 胡希明. Linux 内核源代码分析[M]. 杭州: 浙江大学出版社, 2001.

[5] Texas Instruments. TMS320C6000 CPU and Instruction Set Reference Guide Literature Number: SPRU189F[M]. Texas: Texas Instruments Incorporated, 2000.

[6] Texas Instruments. DSP/ BIOS Driver Development Kit(DDK) Release Version: 1.10[M]. Texas: Texas Instruments Incorporated, 2003.

(上接第 184 页)

拟的准确率为 N'/N 。

根据以上介绍的方法, 以某网站为平台, 将仿真器生成的数据与该网站真实数据进行了验证, 证明是有效的。

3 结束语

得到可靠的访问日志会话对于 Web 使用挖掘与网站结构优化至关重要。文中采用数学建模的方法, 设计并实现了一个网站日志仿真器。它以 Markov 链模型对访问者的行为进行模拟, 得到比较准确的访问日志数据, 使得模拟的会话数据更加准确, 更符合实际。

参考文献:

[1] Kosla R, Blockeel H. Web Mining Research: A Survey[J].

ACM SIG KDD Explorations, 2000, 2(1): 1- 15.

[2] 石磊, 陶永才. Web 对象访问特征模拟器的设计与实现[J]. 计算机仿真, 2006, 23(1): 133- 136.

[3] 杨捷, 毋国庆. 根据用户行为网上导航的方法[J]. 计算机研究与发展, 2005, 42(5): 765- 770.

[4] Brin S, Page L. The Anatomy of a Large Scale Hypertextual Web Search Engine[J]. Computer Networks and ISDN Systems, 1998, 30: 107- 117.

[5] Arasu A, Cho J, Garcia- Molina H, et al. Searching the Web[J]. ACM Transactions on Internet Technology, 2001, 1(1): 2- 43.

[6] Hay B, Wets G, Vanhoof K. Segmentation of Visiting Patterns on Websites Using a Sequence Alignment Method[C] // WE-BKDD 2002- Mining Web Data for Discovering Usage Patterns and Profiles. 4th International Workshop, Revised Papers. Lecture Notes in Artificial Intelligence Vol. 2703. [s. l.]: [s. n.], 2003: 50- 65.